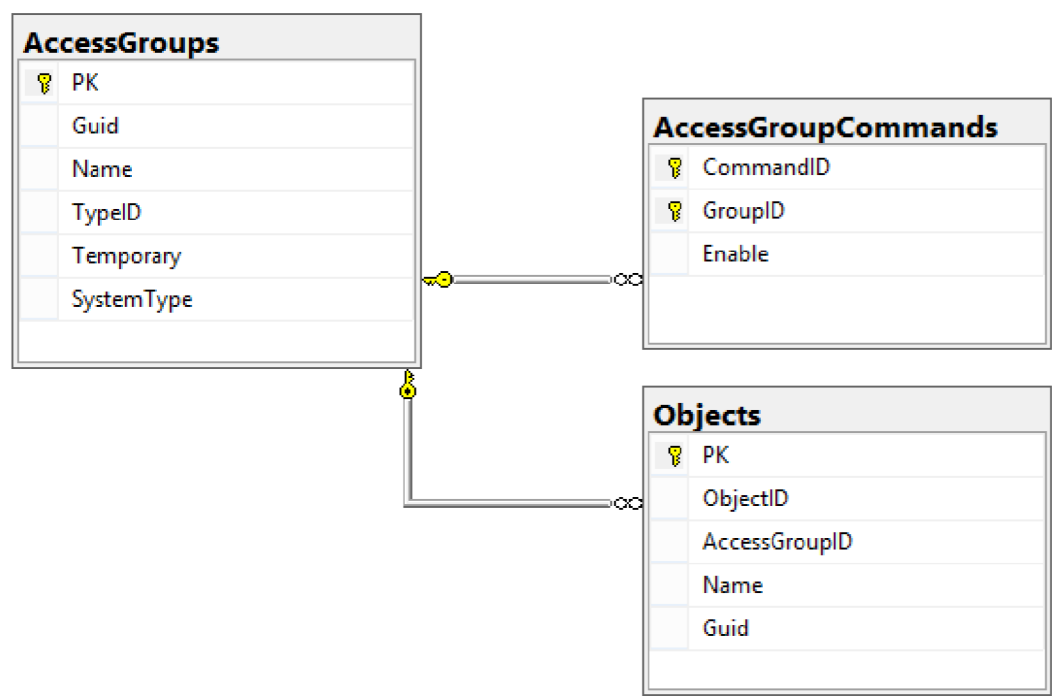


ОСНОВНЫЕ ТЕСТОВЫЕ ЗАДАНИЯ

ИСХОДНЫЕ ДАННЫЕ

База данных **TESTS_SOURCE** содержит три таблицы, в которых хранятся описания групп прав доступа и коды команд, которые разрешены или запрещены для данных групп, а также таблица со списком объектов - **Objects**.

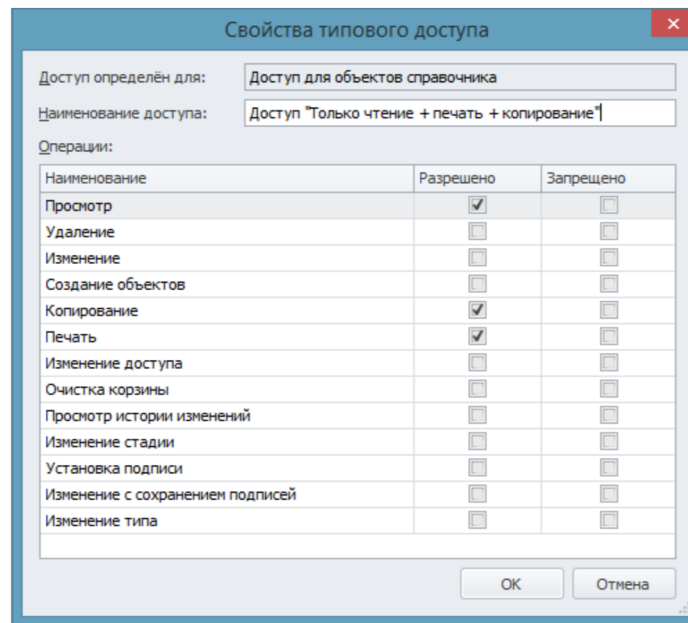


Группы хранятся в таблице **AccessGroups**, допустимые или запрещённые команды – в **AccessGroupCommands**.

- 1. Есть некий стандартный набор групп прав доступа (хранится в **AccessGroups**):

Наименование	Тип
Авторский	Доступ для объектов справочника
Административный	Доступ для объектов справочника
Просмотр	Доступ для объектов справочника
Редакторский	Доступ для объектов справочника
Авторский	Доступ для справочника
Административный	Доступ для справочника
Просмотр	Доступ для справочника
Редакторский	Доступ для справочника
Административный	Доступ для всей системы
Полный	Доступ для связи
Подключение	Доступ для связи
Просмотр	Доступ для связи
Запрет на редактирование	Доступ для объектов справочника
Без ограничений	Доступ для объектов справочника
Пользовательский	Доступ для всей системы
Просмотр и создание	Доступ для объектов справочника
Доступ "Только чтение + печать + копирование"	Доступ для объектов справочника
Доступ "Просмотр + изменение + права доступа"	Доступ для объектов справочника

- 2. Каждая из таких групп содержит перечень разрешённых или запрещённых команд (хранятся в **AccessGroupCommands**, где *GroupID* соответствует *PK* из таблицы **AccessGroups**):



3. Существует некий список объектов, для которых требуется назначить права доступа (**Objects**):

	Наименование	ID	Guid
	Объект 001	1	f2071968-016d-4e6b-bb41-df07d2713dc0
	Объект 002	2	33e66ab2-4d52-4448-a8e6-5c32a3292d12
	Объект 003	3	7ff366b6-705d-41ce-9417-b88dbf103fc5
	Объект 004	4	6cfd2b69-a02e-4299-9eae-5e4e6f52e0e9
	Объект 005	5	bae2e1e2-aff7-4ed2-816e-f767999ac29b

Группа прав доступа *AccessGroupID* таблицы **Objects** соответствует *PK* из таблицы **AccessGroups**.

Предположим, что ответственный за управление доступами назначил каждому объекту не какой-то стандартный набор доступов из списка п.1 (например, «Просмотр»), а для каждого объекта создавал новый набор, в котором указывал разрешённые и запрещённые команды, а затем применял его к этому объекту. В итоге для всех объектов в списке образовалось такое же количество новых групп прав доступа, часть которых дублирует друг друга.

Примечания:

1. У стандартных групп значение колонки *Temporary* равно нулю, у групп, которые были созданы явно для каждого объекта из списка п.3 – равно единице.
2. В приложении №1 находится скрипт, который создаёт две тестовые базы данных (**перед выполнением требуется исправить пути к файлам!**) и заполняет требуемыми исходными данными первую базу.

ТЕСТОВОЕ ЗАДАНИЕ №1

Требуется создать новую базу данных **TEST_DEST**, с идентичными таблицами, с одной оговоркой – в таблице **AccessGroups** значения первичного ключа должны начинаться со значения 1000 и генерироваться с шагом 50. Далее необходимо скопировать информацию о группах прав доступа и их командах, сохранив содержимое групп, а также содержимое таблицы **Objects**, также сохраняя права доступа.

ТЕСТОВОЕ ЗАДАНИЕ №2

После выполнения тестового задания №1 требуется исправить ситуацию с избыточностью групп прав доступа, т.е. заменить все идентичные по командам группы прав доступа, которые были созданы для объектов списка п.3, на первые подходящие группы. При этом требуется удалить лишние группы и их команды из таблиц,

Далее необходимо преобразовать временные группы в постоянные (указав значение колонки *Temporary* равным 0 и задав какое-то название группы вместо пустой строки).

ПРИМЕЧАНИЕ: Перед отправкой результатов на проверку проверьте, выполняются ли ваши скрипты, соответствуют ли результаты работы ваших скриптов условиям заданий.

ПРИЛОЖЕНИЕ №1

/* ==Параметры сценариев==

Версия целевого сервера : SQL Server 2008 R2

Выпуск целевого ядра СУБД : Выпуск Microsoft SQL Server Express Edition

Тип целевого ядра СУБД : Изолированный SQL Server

*/

USE [master]

GO

SET NOCOUNT ON

-- !!! ВНИМАНИЕ !!!

-- !!! Исправить пути к создаваемым базам данных !!!

CREATE DATABASE [TESTS_SOURCE]

ON PRIMARY

(NAME = N'TESTS_SOURCE', FILENAME = N'F:\Databases\Backups\TESTS_SOURCE.mdf' , SIZE = 1024KB , MAXSIZE = 1GB, FILEGROWTH = 1%)

LOG ON

(NAME = N'TESTS_SOURCE_log', FILENAME = N'F:\Databases\Backups\TESTS_SOURCE.ldf' , SIZE = 1024KB , MAXSIZE = 1GB, FILEGROWTH = 1%)

GO

CREATE DATABASE [TESTS_DEST]

ON PRIMARY

(NAME = N'TESTS_DEST', FILENAME = N'F:\Databases\Backups\TESTS_DESTS.mdf' , SIZE = 1024KB , MAXSIZE = 1GB, FILEGROWTH = 1%)

LOG ON

(NAME = N'TESTS_DEST_log', FILENAME = N'F:\Databases\Backups\TESTS_DESTS.ldf' , SIZE = 1024KB , MAXSIZE = 1GB, FILEGROWTH = 1%)

GO

-- [TESTS_SOURCE].[AccessGroups]

USE [TESTS_SOURCE]

GO

CREATE TABLE [dbo].[AccessGroups](

[PK] [int] IDENTITY(1,1) NOT NULL,

[Guid] [uniqueidentifier] NOT NULL,

```

[Name] [nvarchar](255) NOT NULL,

[TypeID] [int] NOT NULL,

[Temporary] [int] NOT NULL,

[SystemType] [int] NOT NULL,

CONSTRAINT [PK_AccessGroups] PRIMARY KEY CLUSTERED
(
    [PK] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
) ON [PRIMARY]
GO

```

```

ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Guid] DEFAULT (newid()) FOR [Guid]
ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Name] DEFAULT (") FOR [Name]
ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_TypeID] DEFAULT ((0)) FOR [TypeID]
ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Temporary] DEFAULT ((0)) FOR [Temporary]
ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_SystemType] DEFAULT ((0)) FOR [SystemType]

```

```

CREATE NONCLUSTERED INDEX [IX_AccessGroups_TypeID] ON [dbo].[AccessGroups](
    [TypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

CREATE UNIQUE NONCLUSTERED INDEX [UK_AccessGroups_Guid] ON [dbo].[AccessGroups](
    [Guid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

```

```
-- [TESTS_SOURCE].[AccessGroupCommands]
```

```
USE [TESTS_SOURCE]
```

```
GO
```

```

CREATE TABLE [dbo].[AccessGroupCommands](
    [CommandID] [int] NOT NULL,
    [GroupID] [int] NOT NULL,
    [Enable] [int] NOT NULL,
    CONSTRAINT [PK_AccessGroupCommands] PRIMARY KEY CLUSTERED
(

```

```

[CommandID] ASC,

[GroupID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]

) ON [PRIMARY]

GO

```

```

CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_CommandID] ON [dbo].[AccessGroupCommands](

[CommandID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_GroupID] ON [dbo].[AccessGroupCommands](

[GroupID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_Optimize1] ON [dbo].[AccessGroupCommands](

[CommandID] ASC,

[GroupID] ASC,

[Enable] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_CommandID] DEFAULT ((0)) FOR [CommandID]

ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_GroupID] DEFAULT ((0)) FOR [GroupID]

ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_Enable] DEFAULT ((0)) FOR [Enable]

ALTER TABLE [dbo].[AccessGroupCommands] WITH CHECK ADD CONSTRAINT [FK_AccGrpCommands_AccGroups] FOREIGN KEY([GroupID])
REFERENCES [dbo].[AccessGroups] ([PK]) ON DELETE CASCADE

ALTER TABLE [dbo].[AccessGroupCommands] CHECK CONSTRAINT [FK_AccGrpCommands_AccGroups]

GO

```

```

-- [TESTS_SOURCE].[Objects]

```

```

USE [TESTS_SOURCE]

GO

```

```

CREATE TABLE [dbo].[Objects](

[PK] [int] IDENTITY(1,1) NOT NULL,

[ObjectID] [int] NOT NULL,

[AccessGroupID] [int] NOT NULL,

```

```

[Name] [nvarchar](255) NOT NULL,

[Guid] [uniqueidentifier] NOT NULL,

CONSTRAINT [PK_Objects] PRIMARY KEY CLUSTERED

(

    [PK] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]

) ON [PRIMARY]

GO

```

```

ALTER TABLE [dbo].[Objects] ADD DEFAULT ((0)) FOR [ObjectID]

ALTER TABLE [dbo].[Objects] ADD DEFAULT ((0)) FOR [AccessGroupID]

ALTER TABLE [dbo].[Objects] ADD DEFAULT (N'') FOR [Name]

ALTER TABLE [dbo].[Objects] ADD CONSTRAINT [DF_Objects_Guid] DEFAULT (newid()) FOR [Guid]

ALTER TABLE [dbo].[Objects] WITH CHECK ADD CONSTRAINT [FK_Objects_AccGroups] FOREIGN KEY([AccessGroupID]) REFERENCES
[dbo].[AccessGroups] ([PK]) ON DELETE SET DEFAULT ON UPDATE CASCADE

ALTER TABLE [dbo].[Objects] CHECK CONSTRAINT [FK_Objects_AccGroups]

GO

```

```

-- [TESTS_DEST].[AccessGroups]

```

```

USE [TESTS_DEST]

GO

```

```

CREATE TABLE [dbo].[AccessGroups](

    [PK] [int] IDENTITY(1000,50) NOT NULL,

    [Guid] [uniqueidentifier] NOT NULL,

    [Name] [nvarchar](255) NOT NULL,

    [TypeID] [int] NOT NULL,

    [Temporary] [int] NOT NULL,

    [SystemType] [int] NOT NULL,

    CONSTRAINT [PK_AccessGroups] PRIMARY KEY CLUSTERED

(

    [PK] ASC

) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]

) ON [PRIMARY]

GO

```

```

ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Guid] DEFAULT (newid()) FOR [Guid]

```

```
ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Name] DEFAULT (") FOR [Name]

ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_TypeID] DEFAULT ((0)) FOR [TypeID]

ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_Temporary] DEFAULT ((0)) FOR [Temporary]

ALTER TABLE [dbo].[AccessGroups] ADD CONSTRAINT [DF_AccessGroups_SystemType] DEFAULT ((0)) FOR [SystemType]
```

```
CREATE NONCLUSTERED INDEX [IX_AccessGroups_TypeID] ON [dbo].[AccessGroups](
    [TypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
CREATE UNIQUE NONCLUSTERED INDEX [UK_AccessGroups_Guid] ON [dbo].[AccessGroups](
    [Guid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

GO

```
-- [TESTS_DEST].[AccessGroupCommands]
```

```
USE [TESTS_DEST]
```

GO

```
CREATE TABLE [dbo].[AccessGroupCommands](
    [CommandID] [int] NOT NULL,
    [GroupID] [int] NOT NULL,
    [Enable] [int] NOT NULL,
    CONSTRAINT [PK_AccessGroupCommands] PRIMARY KEY CLUSTERED
(
    [CommandID] ASC,
    [GroupID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
) ON [PRIMARY]

GO
```

```
CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_CommandID] ON [dbo].[AccessGroupCommands](
    [CommandID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_GroupID] ON [dbo].[AccessGroupCommands](
    [GroupID] ASC
```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IX_AccGroupCommands_Optimize1] ON [dbo].[AccessGroupCommands](  
    [CommandID] ASC,  
    [GroupID] ASC,  
    [Enable] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_CommandID] DEFAULT ((0)) FOR [CommandID]
```

```
ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_GroupID] DEFAULT ((0)) FOR [GroupID]
```

```
ALTER TABLE [dbo].[AccessGroupCommands] ADD CONSTRAINT [DF_AccGrpCommands_Enable] DEFAULT ((0)) FOR [Enable]
```

```
ALTER TABLE [dbo].[AccessGroupCommands] WITH CHECK ADD CONSTRAINT [FK_AccGrpCommands_AccGroups] FOREIGN KEY([GroupID])  
REFERENCES [dbo].[AccessGroups] ([PK]) ON DELETE CASCADE
```

```
ALTER TABLE [dbo].[AccessGroupCommands] CHECK CONSTRAINT [FK_AccGrpCommands_AccGroups]
```

```
GO
```

```
-- [TESTS_DEST].[Objects]
```

```
USE [TESTS_DEST]
```

```
GO
```

```
CREATE TABLE [dbo].[Objects](  
    [PK] [int] IDENTITY(1,1) NOT NULL,  
    [ObjectID] [int] NOT NULL,  
    [AccessGroupID] [int] NOT NULL,  
    [Name] [nvarchar](255) NOT NULL,  
    [Guid] [uniqueidentifier] NOT NULL,  
    CONSTRAINT [PK_Objects] PRIMARY KEY CLUSTERED
```

```
(  
    [PK] ASC  
)  
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[Objects] ADD DEFAULT ((0)) FOR [ObjectID]
```

```
ALTER TABLE [dbo].[Objects] ADD DEFAULT ((0)) FOR [AccessGroupID]
```

```
ALTER TABLE [dbo].[Objects] ADD DEFAULT (N'') FOR [Name]
```

```
ALTER TABLE [dbo].[Objects] ADD CONSTRAINT [DF_Objects_Guid] DEFAULT (newid()) FOR [Guid]
```